VBA Experiment: Building Address Points From Parcels

Contributed by Bert Granberg

05, Aug. 2010
Last Updated 05, Aug. 2010

At worst, building a GIS data set of address points would involve time consuming, inefficient field collection with GPS units.

Luckily there are many existing data resources from which to start building address points that can quickly deliver large quantities of high quality points. Existing resources include tables of known addresses, street centerlines with address ranges, and parcels that carry addresses.

Converting parcels to address points presents several hurdles:
- Many parcels can share the same address
- A parcel can contain several addresses
- Some parcels do not have an address (easements, agricultural and range lands, as examples)
- The centroid is easy to get for a parcel but ideally the address point is the primary building OR at a frontage point near the address's primary access point to the road network.

The VBA code below is a work in progress and attempts to convert parcels to a set of merged 'Address Polygons' wherein the shared boundarys of adjacent polygons with the same address are removed. Next it assigns the parcel a frontage point by determining a point close to the street that carries the same name as the parcel that is within the parcel.

The idea is to go from 1) Raw Parcels --> 2) Address Polygons --> 3) Address Points (in this case frontage points). The data below illustrates the idea using parcels in a neighborhood within SL County.

1. Raw Parcels

2) Address Polygons (shown in red, gray lines indicate parcel boundaries that were removed)

3) Address Points (frontage style)

Next steps include trying to refine the frontage placement on cul de sacs and other curving roads, and, in general, more testing.

If you have any suggestionsor comments on how this automation task might better be accomplished, please email bgranberg@utah.gov.

```
'8/5/2010
'DRAFT CODE, CHECK BACK FOR UPDATES
'STILL NEEDS MORE WORK!


Public Sub createAddressPolysAndPointsFromParcelsWithSitusAddress()

    Debug.Print "Start " & Now

    'Get Reference To Current ArcMap Session
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Set pMxDoc = ThisDocument
    Set pMap = pMxDoc.FocusMap

    Dim pSrcFLayer As IFeatureLayer
Dim pSrcFClass As IFeatureClass
Dim pSrcTable As ITable
Dim pMergePolyFLayer As IFeatureLayer
Dim pMergePolyFClass As IFeatureClass
Dim pMergePtFLayer As IFeatureLayer
Dim pMergePtFClass As IFeatureClass
Dim pRoadFLayer As IFeatureLayer
Dim pRoadFClass As IFeatureClass
Dim pQF As IQueryFilter
Dim parcelAddFName, streetNameFName As String
Dim parcelAddFI, streetNameFI As Long

'*******
'SET THIS: Get the first layer in the table of contents, layer 0
'the featureclass you are using must be referenced by the
'first layer in the map
Set pSrcFLayer = pMap.Layer(2)
Set pMergePolyFLayer = pMap.Layer(1)
Set pMergePtFLayer = pMap.Layer(0)
Set pRoadFLayer = pMap.Layer(3)
parcelAddFName = "PARCEL_ADD"
streetNameFName = "S_NAME"
'*******

Set pSrcFClass = pSrcFLayer.FeatureClass
parcelAddFI = pSrcFClass.FindField(parcelAddFName)
Set pMergePolyFClass = pMergePolyFLayer.FeatureClass
Set pMergePtFClass = pMergePtFLayer.FeatureClass
Set pRoadFClass = pRoadFLayer.FeatureClass
roadnameFI = pRoadFClass.FindField(streetNameFName)
Set pSrcTable = pSrcFClass ' QI
Set pQF = New QueryFilter

'set up tablesort to return results of a multi-field sort
Dim pTableSort As ITableSort
Set pTableSort = New TableSort 'VBA Query Interface
'the fields used and their order or precedence for the sort
pTableSort.Fields = (parcelAddFName)
pTableSort.Ascending(parcelAddFName) = True
Set pTableSort.QueryFilter = pQF

'sort on only selected features
Set pTableSort.Table = pSrcTable

pTableSort.Sort Nothing

'get a cursor to iterate thru the features returned by the tablesort
Dim pFCursor As IFeatureCursor
Dim pFeature As IFeature
Dim pNewFeature As IFeature
Set pFCursor = pTableSort.Rows


Dim addStr As String 'concatenate xy coord of label point with area and perimeter
Dim lastAddStr As String 'track xyAPStr from previous records
Dim Count As Long
Dim pPoly, pNewPoly As IPolygon
Dim pPolyTopOp As ITopologicalOperator
Dim pPt As IPoint
Dim pArea As IArea
Dim compare() As Long
lastAddStr = ""

Set pFeature = pFCursor.NextFeature
lastAddStr = Trim(Split(pFeature.Value(parcelAddFI), "#")(0))
```

```
Set pPoly = New Polygon
Do Until pFeature Is Nothing

    Count = Count + 1

    addStr = Trim(Split(pFeature.Value(parcelAddFI), "#")(0))
    If addStr = "1695 S 1600 E" Then
        Debug.Print "here"
    End If
    If addStr = lastAddStr Then
        'add to geometry
        Set pNewPoly = pFeature.Shape
        Set pPolyTopOp = pPoly
        Set pPoly = pPolyTopOp.Union(pNewPoly)
    Else
        'write new dissolved address polygon
        Set pNewFeature = pMergePolyFClass.CreateFeature
        Set pNewFeature.Shape = pPoly
        pNewFeature.Value(parcelAddFI) = lastAddStr
        pNewFeature.Store

        'write new address point
        Set pPt = generateFrontagePt(pNewFeature, parcelAddFI, roadnameFI, pRoadFClass, True, 50)
        Set pNewFeature = pMergePtFClass.CreateFeature
        Set pArea = pPoly
        If pArea.Centroid.X = pPt.X And pArea.Centroid.Y = pPt.Y Then
            pNewFeature.Value(4) = "Parcel Centroid1"
        ElseIf pArea.LabelPoint.X = pPt.X And pArea.LabelPoint.Y = pPt.Y Then
            pNewFeature.Value(4) = "Parcel Centroid2"
        Else
            pNewFeature.Value(4) = "Parcel Frontage"
        End If
        Set pNewFeature.Shape = pPt
        pNewFeature.Value(2) = lastAddStr
        pNewFeature.Store

        Set pPoly = pFeature.Shape
    End If
    lastAddStr = addStr

    If Count / 1000 = CInt(Count / 1000) Then
        Debug.Print Count & " " & pRow.Value(labelPtFI)
    End If

    Set pFeature = pFCursor.NextFeature
Loop

'for last record
'write new dissolved address polygon
Set pNewFeature = pMergePolyFClass.CreateFeature
Set pNewFeature.Shape = pPoly
pNewFeature.Value(parcelAddFI) = lastAddStr
pNewFeature.Store

'write new address point
Set pPt = generateFrontagePt(pNewFeature, parcelAddFI, roadnameFI, pRoadFClass, True, 50)
Set pNewFeature = pMergePtFClass.CreateFeature
Set pArea = pPoly
If pArea.Centroid.X = pPt.X And pArea.Centroid.Y = pPt.Y Then
    pNewFeature.Value(4) = "Parcel Centroid1"
ElseIf pArea.LabelPoint.X = pPt.X And pArea.LabelPoint.Y = pPt.Y Then
    pNewFeature.Value(4) = "Parcel Centroid2"
Else
    pNewFeature.Value(4) = "Parcel Frontage"
End If
```

```
  Set pNewFeature.Shape = pPt
  pNewFeature.Value(2) = lastAddStr
  pNewFeature.Store
  Debug.Print "Finish " & Now
End Sub

Public Function generateFrontagePt(ByVal inPolyFeature As IFeature, ByVal inParcelAddFI As Integer, ByVal
inRoadNameFI As Integer, inRoadFC As IFeatureClass, doFrontage, frontageSearchRadius) As IPoint
  Dim pArea As IArea
  Dim pRelOp As IRelationalOperator
  Set pArea = inPolyFeature.ShapeCopy
  Set pRelOp = pArea
  Set generateFrontagePt = pArea.Centroid
  If Not pRelOp.Contains(generateFrontagePt) Then
    Set generateFrontagePt = pArea.LabelPoint
  End If
  If doFrontage Then
    Dim pTopOp As ITopologicalOperator
    Dim pProxOp As IProximityOperator
    Dim pFCursor As IFeatureCursor
    Dim pRoadFeature As IFeature
    Dim streetNameQ As String
    Dim pBuffPoly As IPolygon
    Dim pSF As ISpatialFilter
    Dim distance, closestDistance As Double
    Dim closestFeature As IFeature
    Dim pClosestPoint As IPoint
    Dim pLine As ILine

    Set pTopOp = generateFrontagePt
    Set pBuffPoly = pTopOp.Buffer(frontageSearchRadius)
    'simple get street name
    streetName = Split(inPolyFeature.Value(inParcelAddFI), " ")(1)
    If Len(streetName) = 1 Then
      streetName = Split(inPolyFeature.Value(inParcelAddFI), " ")(2)
    End If
    Set pSF = New SpatialFilter
    pSF.WhereClause = pRoadFeature.Fields.Field(inRoadNameFI).Name & " like '%" & streetName & "%'"
    Set pSF.Geometry = pBuffPoly
    pSF.SpatialRel = esriSpatialRelIntersects
    Set pFCursor = inRoadFC.Search(pSF, True)
    Set pRoadFeature = pFCursor.NextFeature

    closestDistance = frontageSearchRadius + 100
    Do Until pRoadFeature Is Nothing
      Set pProxOp = pRoadFeature.ShapeCopy
      distance = pProxOp.ReturnDistance(generateFrontagePt)
      If distance < closestDistance Then
        pclosestFeature = pRoadFeature
        Set pClosestPoint = pProxOp.ReturnNearestPoint(generateFrontagePt, esriNoExtension)
        closestDistance = distance
      End If
      Set pRoadFeature = pFCursor.NextFeature
    Loop

    If Not pClosestPoint Is Nothing Then
      Set pProxOp = inPolyFeature.ShapeCopy
      Set pClosestPoint = pProxOp.ReturnNearestPoint(pClosestPoint, esriNoExtension)
      Set pLine = New Line
      pLine.PutCoords pClosestPoint, generateFrontagePt
      pLine.QueryPoint esriNoExtension, 5, False, generateFrontagePt
    End If

  End If
End Function
```